Package 'modelsummary'

June 16, 2020

Type Package

Title Summary Tables for Statistical Models: Beautiful, Customizable, and Publication-Ready

Description Create beautiful and customizable tables to summarize several statistical models side-by-side. This package supports dozens of model types and can produce tables in HTML, LaTeX, Markdown, Word, PowerPoint, Excel, RTF, JPG, or PNG. Tables can easily be embedded in 'Rmarkdown' or 'knitr' dynamic documents.

Version 0.4.1

URL https://vincentarelbundock.github.io/modelsummary

BugReports https://github.com/vincentarelbundock/modelsummary/issues

Depends R (>= 3.4.0)

Imports broom,
checkmate,
dplyr,
generics,
gt (>= 0.2.0),
knitr (>= 1.16),
magrittr,
kableExtra,
purrr,
rmarkdown (>= 1.6.0),
stringr,
tibble,
tidyr (>= 1.0.0)

Suggests broom.mixed,
flextable,
huxtable,
lmtest,
MASS,
officer,
sandwich,
testthat

License GPL-3

Encoding UTF-8

LazyData false

RoxygenNote 7.1.0.9000

R topics documented:

`clean_latex` *Deprecated function*

Description

The 'gt::as_latex' function does not (yet) produce compilable LaTeX code. This function used to clean up LaTeX output to allow compilation to PDF. This function is now deprecated since 'modelsummary' currently supports 'kableExtra', which has a mature LaTeX rendering engine.

Usage

```
clean_latex(...)
```

Arguments

`...`	catch everything

`extract` *Extract and combine estimates and goodness-of-fit statistics from several statistical models.*

Description

Extract and combine estimates and goodness-of-fit statistics from several statistical models.

Usage

```
extract(
  models,
  statistic = "std.error",
  statistic_override = NULL,
  statistic_vertical = TRUE,
  conf_level = 0.95,
  coef_map = NULL,
  coef_omit = NULL,
  gof_map = modelsummary::gof_map,
  gof_omit = NULL,
```

```
  add_rows = NULL,
  stars = FALSE,
  fmt = "%.3f",
  estimate = "estimate",
  add_rows_location = NULL,
  ...
)
```

Arguments

models	a single model object or a (potentially named) list of models to summarize
statistic	string name of the statistic to include in parentheses • Typical values: "conf.int", "std.error", "statistic", "p.value" • Alternative values: any column name produced by ‘broom::tidy(model)‘
statistic_override	manually override the uncertainy estimates. This argument accepts three types of input: • a function or list of functions of length(models) which produce variance-covariance matrices with row and column names equal to the names of your coefficient estimates. For example, ‘R‘ supplies the ‘vcov‘ function, and the ‘sandwich‘ package supplies ‘vcovHC‘, ‘vcovHAC‘, etc. • a list of length(models) variance-covariance matrices with row and column names equal to the names of your coefficient estimates. • a list of length(models) vectors with names equal to the names of your coefficient estimates. Numeric vectors are formatted according to ‘fmt‘ and placed in brackets, character vectors printed as given.
statistic_vertical	TRUE if statistics should be printed below estimates. FALSE if statistics should be printed beside estimates.
conf_level	confidence level to use for confidence intervals
coef_map	named character vector. Names refer to the original variable names. Values refer to the variable names that will appear in the table. Coefficients which are omitted from this vector will be omitted from the table. The table will be ordered in the same order as this vector.
coef_omit	string regular expression. Omits all matching coefficients from the table (using ‘stringr::str_detect‘).
gof_map	data.frame with four columns: ‘raw‘, ‘clean‘, ‘fmt‘, and ‘omit‘. See ‘modelsummary::gof_map‘
gof_omit	string regular expression. Omits all matching gof statistics from the table (using ‘stringr::str_detect‘).
add_rows	a data.frame (or tibble) with the following columns: • section (character): insert in "middle" or "bottom" section of the table • position (integer): row position in the section • term (character): string to display under coefficient names • one column per model with the same name as that model with the values to insert (some models can be omitted). • See the examples section of this documentation and an example.
stars	to indicate statistical significance

- FALSE (default): no significance stars.
- TRUE: *=.1, **=.05, ***=.01
- Named numeric vector for custom stars such as 'c('*' = .1, '+' = .05)'

`fmt` string which specifies how numeric values will be rounded. This string is passed to the 'sprintf' function. '%.3f' will keep 3 digits after the decimal point with trailing zero. '%.5f' will keep 5 digits. '%.3e' will use exponential notation. See '?sprintf' for more options.

`estimate` character name of the estimate to display. Must be a column name in the dataframe produced by 'tidy(model)'. In the vast majority of cases, the default value of this argument should not be changed.

`add_rows_location` This argument is deprecated. Use a data.frame as described in the documentation for the 'add_rows' argument.

`...` all other arguments are passed to the 'tidy' method used to extract estimates from the model. For example, this allows users to set 'exponentiate=TRUE' to exponentiate logistic regression coefficients.

Value

tibble

Examples

```
library(modelsummary)
data(trees)
models <- list()
models[['Bivariate']] <- lm(Girth ~ Height, data = trees)
models[['Multivariate']] <- lm(Girth ~ Height + Volume, data = trees)
extract(models)
```

---

glance_custom *Extract custom information from a model object and turn it into a tidy tibble with a single row.*

Description

glance_custom methods always return either a one-row data frame (except on 'NULL', which returns an empty data frame). This

Usage

```
glance_custom(x)
```

Arguments

`x` model or other R object to convert to single-row data frame

Methods

No methods found in currently loaded packages.

glance_more *Glance *more* at an object*

Description

Construct a single row summary "glance" of a model, fit, or other object. This method extracts more information than the usual ‘glance‘ method. Defining a ‘glance_more‘ allows users to customize the features of a model to display in a ‘modelsummary‘ table.

Usage

```
glance_more(x, ...)
```

Arguments

x	model or other R object to convert to single-row data frame
...	other arguments passed to methods

Details

glance_more methods must always return a one-row data frame

gof_map *Data.frame used to clean up and format goodness-of-fit statistics*

Description

By default, this data frame is passed to the 'gof_map' argument of the 'msummary' or 'modelsummary' functions. Users can modify this data frame to customize the list of statistics to display and their format. See example below.

Usage

```
gof_map
```

Format

data.frame with 4 columns of character data: raw, clean, fmt, omit

Examples

```
library(modelsummary)
mod <- lm(wt ~ drat, data = mtcars)
gm <- modelsummary::gof_map
gm$omit[gm$raw == 'deviance'] <- FALSE
gm$fmt[gm$raw == 'r.squared'] <- "%.5f"
msummary(mod, gof_map = gm)
```

knit_latex *Deprecated function*

Description

The 'gt::as_latex' function does not (yet) produce compilable LaTeX code. This function used to clean up LaTeX output to allow compilation to PDF. This function is now deprecated since 'modelsummary' currently supports 'kableExtra', which has a mature LaTeX rendering engine.

Usage

```
knit_latex(...)
```

Arguments

...	catch everything

modelsummary *Beautiful, customizable summaries of statistical models*

Description

Beautiful, customizable summaries of statistical models

Usage

```
modelsummary(
  models,
  output = "default",
  fmt = "%.3f",
  statistic = "std.error",
  statistic_override = NULL,
  statistic_vertical = TRUE,
  conf_level = 0.95,
  stars = FALSE,
  coef_map = NULL,
  coef_omit = NULL,
  gof_map = modelsummary::gof_map,
  gof_omit = NULL,
  add_rows = NULL,
  title = NULL,
  notes = NULL,
  estimate = "estimate",
  filename = NULL,
  subtitle = NULL,
  add_rows_location = NULL,
  ...
)
```

Arguments

`models` a single model object or a (potentially named) list of models to summarize

`output` filename or object type (string)

- Supported filename extensions: .html, .tex, .md, .txt, .png, .jpg.
- Supported object types: "default", "html", "markdown", "latex", "gt", "kableExtra", "huxtable", "flextable".
- When a file name is supplied to the ‘output‘ argument, the table is written immediately to file. If you want to customize your table by post-processing it with functions provided by the ‘gt‘ or ‘kableExtra‘ packages, you need to choose a different output format (e.g., "gt", "latex", "html", "markdown"), and you need to save the table after post-processing using the ‘gt::gtsave‘, ‘kable::save_kable‘, or ‘cat‘ functions.

`fmt` string which specifies how numeric values will be rounded. This string is passed to the ‘sprintf‘ function. ’%.3f’ will keep 3 digits after the decimal point with trailing zero. ’%.5f’ will keep 5 digits. ’%.3e’ will use exponential notation. See ‘?sprintf‘ for more options.

`statistic` string name of the statistic to include in parentheses

- Typical values: "conf.int", "std.error", "statistic", "p.value"
- Alternative values: any column name produced by ‘broom::tidy(model)‘

`statistic_override`

manually override the uncertainy estimates. This argument accepts three types of input:

- a function or list of functions of length(models) which produce variance-covariance matrices with row and column names equal to the names of your coefficient estimates. For example, ‘R‘ supplies the ‘vcov‘ function, and the ‘sandwich‘ package supplies ‘vcovHC‘, ‘vcovHAC‘, etc.
- a list of length(models) variance-covariance matrices with row and column names equal to the names of your coefficient estimates.
- a list of length(models) vectors with names equal to the names of your coefficient estimates. Numeric vectors are formatted according to ‘fmt‘ and placed in brackets, character vectors printed as given.

`statistic_vertical`

TRUE if statistics should be printed below estimates. FALSE if statistics should be printed beside estimates.

`conf_level` confidence level to use for confidence intervals

`stars` to indicate statistical significance

- FALSE (default): no significance stars.
- TRUE: *=.1, **=.05, ***=.01
- Named numeric vector for custom stars such as ‘c(’*’ = .1, ’+’ = .05)‘

`coef_map` named character vector. Names refer to the original variable names. Values refer to the variable names that will appear in the table. Coefficients which are omitted from this vector will be omitted from the table. The table will be ordered in the same order as this vector.

`coef_omit` string regular expression. Omits all matching coefficients from the table (using ‘stringr::str_detect‘).

`gof_map` data.frame with four columns: ‘raw‘, ‘clean‘, ‘fmt‘, and ‘omit‘. See ‘modelsummary::gof_map‘

gof_omit	string regular expression. Omits all matching gof statistics from the table (using 'stringr::str_detect').
add_rows	a data.frame (or tibble) with the following columns: • section (character): insert in "middle" or "bottom" section of the table • position (integer): row position in the section • term (character): string to display under coefficient names • one column per model with the same name as that model with the values to insert (some models can be omitted). • See the examples section of this documentation and an example.
title	string
notes	list or vector of notes to append to the bottom of the table.
estimate	character name of the estimate to display. Must be a column name in the dataframe produced by 'tidy(model)'. In the vast majority of cases, the default value of this argument should not be changed.
filename	This argument was deprecated in favor of the 'output' argument.
subtitle	This argument is deprecated. Use 'title' or the 'tab_header'
add_rows_location	This argument is deprecated. Use a data.frame as described in the documentation for the 'add_rows' argument.
...	all other arguments are passed to the 'tidy' method used to extract estimates from the model. For example, this allows users to set 'exponentiate=TRUE' to exponentiate logistic regression coefficients.

Value

a 'gt' table object.

Examples

```
library(modelsummary)

# load data and estimate models
data(trees)
models <- list()
models[['Bivariate']] <- lm(Girth ~ Height, data = trees)
models[['Multivariate']] <- lm(Girth ~ Height + Volume, data = trees)

# simple table
msummary(models)

# confidence intervals, p values, or t-stats instead of standard errors
msummary(models, statistic = 'conf.int', conf_level = 0.99)
msummary(models, statistic = 'p.value', conf_level = 0.99)
msummary(models, statistic = 'statistic', conf_level = 0.99)

# rename and re-order coefficients
msummary(models, coef_map = c('Volume' = 'Large', 'Height' = 'Tall'))

# titles
msummary(models, title = 'This is the title')

# title with italicized text
```

```
msummary(models, title = gt::md('This is *the* title'))

# add_rows: we use `tribble` from the `tibble` package to build a data.frame
# more easily
rows <- tibble::tribble(~term, ~section, ~position, ~Bivariate, ~Multivariate,
                        'Empty row', 'middle', 3, '-', '-',
                        'Another empty row', 'bottom', 1, '?', '?')
msummary(models, add_rows = rows)

# notes at the bottom of the table (here, the second note includes markdown bold characters)
msummary(models, notes = list('A first note', gt::md('A **bold** note')))

# modify list of GOF statistics and their format using the built-in
# 'gof_map' data frame as a starting point
gof_custom <- modelsummary::gof_map
gof_custom$omit[gof_custom$raw == 'deviance'] <- FALSE
gof_custom$fmt[gof_custom$raw == 'r.squared'] <- "%.5f"
msummary(models, gof_map = gof_custom)
```

`msummary` *Beautiful, customizable summaries of statistical models*

Description

'msummary()' is a shortcut to 'modelsummary()'

Usage

```
msummary(
  models,
  output = "default",
  fmt = "%.3f",
  statistic = "std.error",
  statistic_override = NULL,
  statistic_vertical = TRUE,
  conf_level = 0.95,
  stars = FALSE,
  coef_map = NULL,
  coef_omit = NULL,
  gof_map = modelsummary::gof_map,
  gof_omit = NULL,
  add_rows = NULL,
  title = NULL,
  notes = NULL,
  estimate = "estimate",
  filename = NULL,
  subtitle = NULL,
  add_rows_location = NULL,
  ...
)
```

Arguments

models	a single model object or a (potentially named) list of models to summarize
output	filename or object type (string) • Supported filename extensions: .html, .tex, .md, .txt, .png, .jpg. • Supported object types: "default", "html", "markdown", "latex", "gt", "kableExtra", "huxtable", "flextable". • When a file name is supplied to the 'output' argument, the table is written immediately to file. If you want to customize your table by post-processing it with functions provided by the 'gt' or 'kableExtra' packages, you need to choose a different output format (e.g., "gt", "latex", "html", "markdown"), and you need to save the table after post-processing using the 'gt::gtsave', 'kable::save_kable', or 'cat' functions.
fmt	string which specifies how numeric values will be rounded. This string is passed to the 'sprintf' function. '%.3f' will keep 3 digits after the decimal point with trailing zero. '%.5f' will keep 5 digits. '%.3e' will use exponential notation. See '?sprintf' for more options.
statistic	string name of the statistic to include in parentheses • Typical values: "conf.int", "std.error", "statistic", "p.value" • Alternative values: any column name produced by 'broom::tidy(model)'
statistic_override	manually override the uncertainy estimates. This argument accepts three types of input: • a function or list of functions of length(models) which produce variance-covariance matrices with row and column names equal to the names of your coefficient estimates. For example, 'R' supplies the 'vcov' function, and the 'sandwich' package supplies 'vcovHC', 'vcovHAC', etc. • a list of length(models) variance-covariance matrices with row and column names equal to the names of your coefficient estimates. • a list of length(models) vectors with names equal to the names of your coefficient estimates. Numeric vectors are formatted according to 'fmt' and placed in brackets, character vectors printed as given.
statistic_vertical	TRUE if statistics should be printed below estimates. FALSE if statistics should be printed beside estimates.
conf_level	confidence level to use for confidence intervals
stars	to indicate statistical significance • FALSE (default): no significance stars. • TRUE: *=.1, **=.05, ***=.01 • Named numeric vector for custom stars such as 'c('*' = .1, '+' = .05)'
coef_map	named character vector. Names refer to the original variable names. Values refer to the variable names that will appear in the table. Coefficients which are omitted from this vector will be omitted from the table. The table will be ordered in the same order as this vector.
coef_omit	string regular expression. Omits all matching coefficients from the table (using 'stringr::str_detect').
gof_map	data.frame with four columns: 'raw', 'clean', 'fmt', and 'omit'. See 'modelsummary::gof_map'

`gof_omit` string regular expression. Omits all matching gof statistics from the table (using ‘stringr::str_detect‘).

`add_rows` a data.frame (or tibble) with the following columns:

- section (character): insert in "middle" or "bottom" section of the table
- position (integer): row position in the section
- term (character): string to display under coefficient names
- one column per model with the same name as that model with the values to insert (some models can be omitted).
- See the examples section of this documentation and an example.

`title` string

`notes` list or vector of notes to append to the bottom of the table.

`estimate` character name of the estimate to display. Must be a column name in the dataframe produced by ‘tidy(model)‘. In the vast majority of cases, the default value of this argument should not be changed.

`filename` This argument was deprecated in favor of the ‘output‘ argument.

`subtitle` This argument is deprecated. Use ‘title‘ or the ‘tab_header‘

`add_rows_location` This argument is deprecated. Use a data.frame as described in the documentation for the ‘add_rows‘ argument.

`...` all other arguments are passed to the ‘tidy‘ method used to extract estimates from the model. For example, this allows users to set ‘exponentiate=TRUE‘ to exponentiate logistic regression coefficients.

Value

a 'gt' table object.

Examples

```
library(modelsummary)

# load data and estimate models
data(trees)
models <- list()
models[['Bivariate']] <- lm(Girth ~ Height, data = trees)
models[['Multivariate']] <- lm(Girth ~ Height + Volume, data = trees)

# simple table
msummary(models)

# confidence intervals, p values, or t-stats instead of standard errors
msummary(models, statistic = 'conf.int', conf_level = 0.99)
msummary(models, statistic = 'p.value', conf_level = 0.99)
msummary(models, statistic = 'statistic', conf_level = 0.99)

# rename and re-order coefficients
msummary(models, coef_map = c('Volume' = 'Large', 'Height' = 'Tall'))

# titles
msummary(models, title = 'This is the title')

# title with italicized text
```

```
msummary(models, title = gt::md('This is *the* title'))

# add_rows: we use `tribble` from the `tibble` package to build a data.frame
# more easily
rows <- tibble::tribble(~term, ~section, ~position, ~Bivariate, ~Multivariate,
                        'Empty row', 'middle', 3, '-', '-',
                        'Another empty row', 'bottom', 1, '?', '?')
msummary(models, add_rows = rows)

# notes at the bottom of the table (here, the second note includes markdown bold characters)
msummary(models, notes = list('A first note', gt::md('A **bold** note')))

# modify list of GOF statistics and their format using the built-in
# 'gof_map' data frame as a starting point
gof_custom <- modelsummary::gof_map
gof_custom$omit[gof_custom$raw == 'deviance'] <- FALSE
gof_custom$fmt[gof_custom$raw == 'r.squared'] <- "%.5f"
msummary(models, gof_map = gof_custom)
```

`tidy_custom.default` *Extract custom information from a model object and turn it into a tidy tibble*

Description

Extract custom information from a model object and turn it into a tidy tibble

Usage

```
## Default S3 method:
tidy_custom(x)
```

Arguments

`x`	An object to be converted into a tidy [tibble::tibble()].

Value

A [tibble::tibble()] with information about model components.

Index